

Laboratorio Informático  
Departamento Ciencias Básicas  
Universidad Tecnológica Nacional  
Facultad Regional Rosario

# MATHEMATICA

## MANUAL DE INSTRUCCIONES

1º parte: Graficación

Ing. Sara E. De Federico


2010

## Introducción

El programa Mathematica es una herramienta de software de cálculo numérico, simbólico y gráfico que además de ser utilizada por profesionales de diversas áreas, es un excelente facilitador del aprendizaje de las ciencias formales como Matemática, Física, Química, Probabilidad y Estadística, etc. debido a su entorno amigable de edición, y a su ductilidad de programación.

Para una buena y eficaz utilización de este software, es necesario conocer un pequeño conjunto de reglas sintácticas y convenciones de codificación y construcción de sentencias. Las mismas se irán indicando a medida que se avance en los temas.

## Nuevas Versiones

Este Manual fue construido para el Mathematica v. 5, con anotaciones para las versiones 6 y 7. Para una rápida ubicación estas anotaciones son señalizadas con la viñeta: 

## Entorno de trabajo

El Mathematica tiene una hoja de trabajo similar a un editor de texto, en general con fondo blanco. En la parte superior de la ventana, como casi cualquier aplicación bajo Windows tiene una barra de menús.

Para ingresar una sentencia de ejecución, se debe tipear la misma en la hoja de trabajo. Si al llegar al final de la línea se aprieta Enter, el cursor baja a la línea siguiente igual que en un editor de texto, y el programa considera que lo escrito es **parte de la misma sentencia**, a no ser que se haya incluido el punto y coma (;)

Cada sentencia (de uno o varios renglones) está reunida en un bloque llamado **Celda** o **Cell** delimitado a la derecha de la hoja de edición con una marca llamada **Bracket**. Si estando en una celda se pulsa la tecla del cursor ↓, el programa sale de esa sentencia **sin ejecutarla**, marcando una línea divisoria. Al escribir nuevamente, se abre otra celda diferente, marcada con un nuevo bracket.

El Mathematica cuenta con un algoritmo procesador principal que es llamado **Kernel o núcleo**. El mismo se abre al ingresar la primera función y es el primero que procesa las instrucciones. Aparte del kernel, el Mathematica cuenta con miles de **paquetes**, que son librerías que procesan diferentes temas de aplicación más específica, y pueden ser llamados desde el entorno de trabajo.

Para ejecutar una celda se deben oprimir la combinación de teclas **Shift+Enter** ESTANDO EL CURSOR DENTRO DE LA SENTENCIA o con el botón Intro en el teclado numérico. El bracket cambia de color y de forma, tornándose doble, lo que indica que el kernel está evaluando dicha instrucción. Cuando termina de evaluar, el bracket vuelve a su estado original y el programa muestra la salida. También se puede evaluar:

En la versión 5 seleccionando la opción del menú *Kernel*→*Evaluate*→*Evaluate Cells*.

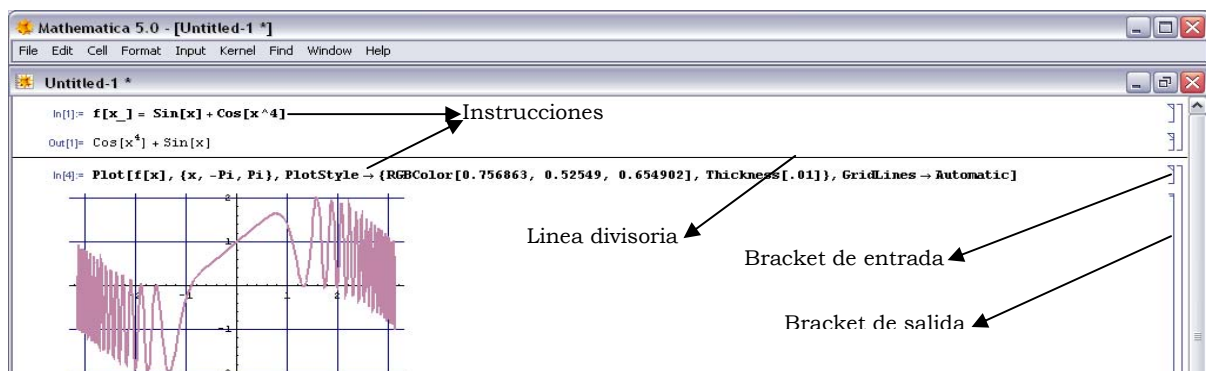


En las versiones 6 y 7 en el menú *Evaluation*→*Evaluate Cells*

Si se seleccionan con el Mouse varios brackets y se selecciona la opción antes descripta, se evalúan todas las celdas a la vez.

La sintaxis de las instrucciones tiene como base el lenguaje en que está construido el kernel, que es el lenguaje C. Se debe respetar estrictamente la simbología.

Pantalla de la versión 5 (el color del menú depende del sistema):





## Versión 6 y 7 (el color del menú depende del sistema):



A diferencia de la versión 5, la barra de menús contiene dos menús nuevos: Graphics y Palettes, y desaparece el menú Kernel.

### Consideraciones importantes acerca del Mathematica

Se deben tener muy en cuenta a la hora de trabajar con el Mathematica las siguientes observaciones:

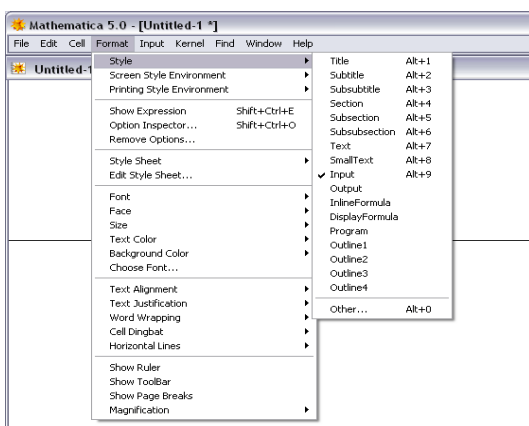
- El lenguaje en que está codificado el Mathematica distingue mayúsculas de minúsculas, por lo tanto no es lo mismo por ejemplo “P” que “p”.
- Cuando se desea nombrar alguna instrucción o ingresar una variable, respetar las convenciones de nombres en programación:
  - ▣ No usar símbolos ni caracteres especiales (%/, &, @, |, ~, etc.)
  - ▣ No usar espacios en blanco (no se puede colocar una frase)
  - ▣ No usar guión alto. Si se puede utilizar guión bajo (\_) para colocar una frase unida
  - ▣ No usar letras mayúsculas, para evitar el uso de letras especiales del programa (se describirán más adelante)
  - ▣ No comenzar el nombre con un número (123funcion NO, funcion123 Sí)
  - ▣ Si bien el uso de acentos está incluido en las últimas versiones del Mathematica, se recomienda no usarlos en nombres.
- Los operadores aritméticos elementales son los mismos de una calculadora CON EXCEPCIÓN DEL MULTIPLICAR, que es reemplazado por un **espacio en blanco**, NO un asterisco, cruz etc. Cuando se expresa la multiplicación de dos elementos (funciones, variables, etc.) los mismos deben ir separados por el espacio en blanco, sino el Mathematica considera a la unión de los dos nombres como uno solo diferente. Por ejemplo la multiplicación de dos variables, x e y se ingresa **x y** y no **xy**, ya que el Mathematica a **xy** lo supone otra variable más, distinta de las anteriores.

### Comandos e instrucciones

Los nombres de los comandos del Mathematica están escritos en **Inglés**, comienzan siempre con **mayúsculas**, y si el nombre de un comando está formado con una combinación de palabras, cada una de ellas comienza con mayúscula. Por ejemplo:

```
Plot[f[x], {x, -10, 10}, PlotStyle->RGBColor[0, 0, 1]];
```

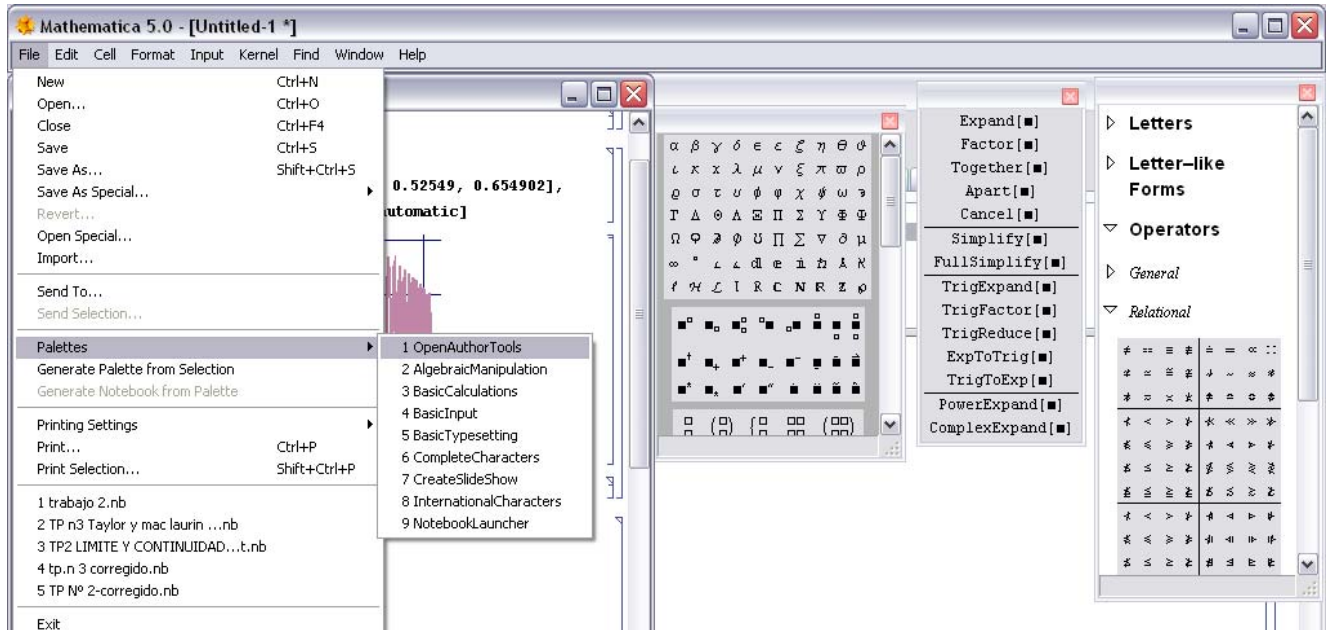
Todo comando del Mathematica va acompañado de un par de corchetes, dentro de los cuales se ingresan los **opciones del mismo**, que indican a quien se aplica este comando y en que formato será la salida. Los opciones a su vez pueden ser también comandos y tener su par de corchetes también.



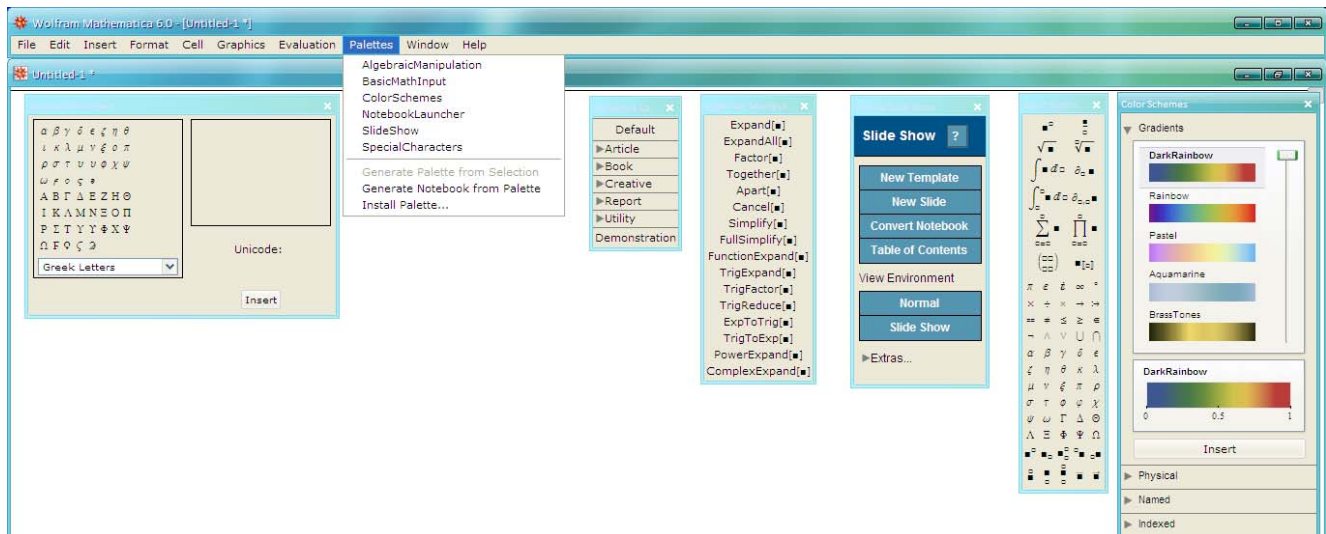
La hoja de trabajo tiene varios modos de texto. Cuando se está ingresando una instrucción, la misma se escribe en letra Courier New negrita tamaño 10. La salida la genera el software en Courier New Normal 10. Los diferentes formatos del texto se muestran en el menú **Format-> Style** en todas las versiones (5,6, y 7).

La sintaxis del Mathematica es estricta tal como se aprecia en el ejemplo, y por ello para facilitar la edición de las instrucciones los nombres de los comandos, la simbología específica, los caracteres especiales, la simbología matemática, y otros elementos para enriquecer y destacar la edición, se pueden seleccionar en plantillas que el programa provee y se encuentran en:

v. 5: el Menú *File*->*Palettes*



v. 6 y 7: menú *Palettes* está directamente en la barra de menús y no contiene exactamente los mismos palettes que en la versión anterior, sino que están agregados palettes de graficación y edición de documentos.



### Definición de funciones

El Mathematica posee las llamadas **Built – In Functions**, en donde tiene predefinidas una profusa cantidad de funciones, clasificadas por tema. Dentro de las **Mathematical Functions** están definidas las funciones más conocidas, por ejemplo: **Sin[*var*]**, **Cos[*var*]**, **Log[*var*]**, **Exp[*var*]**, **Sqrt[*var*]**, **Power[*var*]**. *Var*: variable.

Es importante destacar que este software posee una Ayuda (**Help**) completa, presente en la barra de menús, en donde explica cada comando, sus posibles usos y los acompaña de ejemplos aclaratorios. Además el Help incluye el libro "**Mathematica**" de Stephen Wolfram, creador del concepto inicial de este programa.

Para acceder rápidamente a la Ayuda de algún comando se puede utilizar el operador: `?`  seguido inmediatamente del nombre del comando, la salida es una descripción somera del comando. Para una descripción más completa del mismo, se puede utilizar el operador `??`  seguido del nombre del comando.

Para definir una función, debemos crearla dándole un nombre que por convención comienza con minúsculas y con un carácter alfabético. Además se debe indicar cual es la variable independiente de dicha función. Esto se hace colocando la o las variables independientes entre corchetes a continuación del nombre de la función a crear, seguidos de un guión bajo. Por ejemplo:

```
f[x_]= x^2 + x + 1
```

```
g[x_,y_]= 4 x^2 y + 3 y^5
```

**Nota:** Si se desea evitar la salida de la definición de la función, se puede colocar el símbolo ";", éste oculta la salida de cualquier instrucción de expresiones.

También se puede definir una función con opciones que no son variables independientes, éste elemento también se pone entre corchetes, pero sin el guión bajo:

```
h[x_,n]= x^n + n x
```

```
n x + x^n
```

```
h[x,n]/.n->3
```

```
3 x + x^3
```

En esta función se define un polinomio genérico en donde n se puede reemplazar por cualquier n° o expresión. El símbolo /. significa "tal que" y la flecha → le asigna un valor a n, por lo tanto la expresión significa : "mostrar h[x,n] tal que n= 3".

**Nota:** la flecha se puede hacer con el guión medio y a continuación el signo "mayor que", o sino obtenerla de los Palettes.

El ingreso de una función es evaluado inmediatamente por el núcleo del programa, si hay un error de sintaxis, éste es analizado y sale un mensaje de error. Si en cambio se coloca el símbolo de ":" antes del igual, la instrucción no es evaluada sino hasta que es llamada en otra instrucción:

```
m[x_] := 15 x^2 + 7 + Sen[x]
```

Aquí observamos como se puede analizar una función en un punto, colocando el valor como parámetro entre los corchetes:

```
m[5]
```

```
382+Sen[5]
```

El comando Sen[x] no existe, porque el Mathematica está escrito en inglés, el seno como comando se escribe Sin[x].

```
Clear[m];
```

```
m[x_] := 15 x^2+7 + Sin[x];
```

```
m[5]
```

```
382+Sin[5]
```

**Nota:** En este ejemplo se introduce el comando **Clear[expr]** que limpia todo lugar de memoria del núcleo con el o los nombres que se presentan entre los corchetes, y no genera ninguna salida. Observamos que se ingresan 3 instrucciones en una misma celda, cada instrucción es separada por un ; excepto la última, que es la única que muestra su salida resultante.

En el ejemplo anterior vemos que las salidas de la función  $m[5]$  no son una expresión numérica, esto se debe a que el Mathematica solamente muestra resultados exactos. Para forzar una salida numérica aproximada existe el comando **N[expr]**.

*Expr*: Expresión.

**N[m[5]]**

381.041

Aquí el comando N tiene como parámetro a la función m evaluada en el punto 5. El comando muestra por defecto hasta 6 decimales. Si se desean un número específico de decimales, se coloca como segunda opción:

**N[E^5,70]**

148.4131591025766034211155800405522796234876675938789890467528451109121

Se ha ingresado el valor del número “e” elevado a la 5ª potencia, con 70 decimales.

Como el E, existen varias letras y palabras especiales que definen números específicos en Matemática, todas comienzan con mayúsculas. Otras son:

- **I**: parte imaginaria de un número complejo
- **Pi**: número  $\pi$ .
- **Infinity, ±Infinity**:  $(\infty, \pm\infty)$

El Mathematica permite el uso de comandos, instrucciones y otras expresiones como opciones de otros comandos. Por ejemplo:

**Clear[f,g]**

**f[x\_]=Sqrt[x]**

$\sqrt{x}$

**g[x\_]=x^2 + x**

$x + x^2$

**p[x\_]=Floor[g[x]]**

$\text{Floor}[x + x^2]$

**l[x\_]=f[g[x]]**

$\sqrt{x + x^2}$

**j[x\_]=g[f[x]]**

$\sqrt{x} + x$

Esta capacidad permite la definición de funciones compuestas.

**Floor**: es la parte entera menor o igual a x.

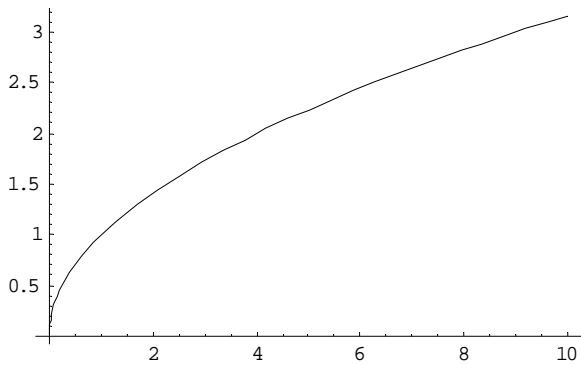
**Sqrt**: raíz cuadrada.

Estas funciones ingresadas se utilizan posteriormente para graficarlas.

## Graficación

Una de las potencialidades del Mathematica es su facultad de generar gráficas de alta resolución, con gran cantidad de opciones. Se puede observar una gráfica de función conocida como no, en un rango específico de valores de la variable independiente, con diferentes niveles de acercamiento, y amplias posibilidades de presentación y coloración.

**Plot[f[x], {x, 0, 10}];**



En el comando **Plot** se debe colocar la función que se desea graficar, y el dominio de graficación, o sea el conjunto de valores que tomará la variable independiente. Luego existen una gran cantidad de opciones que definen calidad, color y elementos adicionales de salida de graficación y que se describen entre los corchetes separados por una coma.

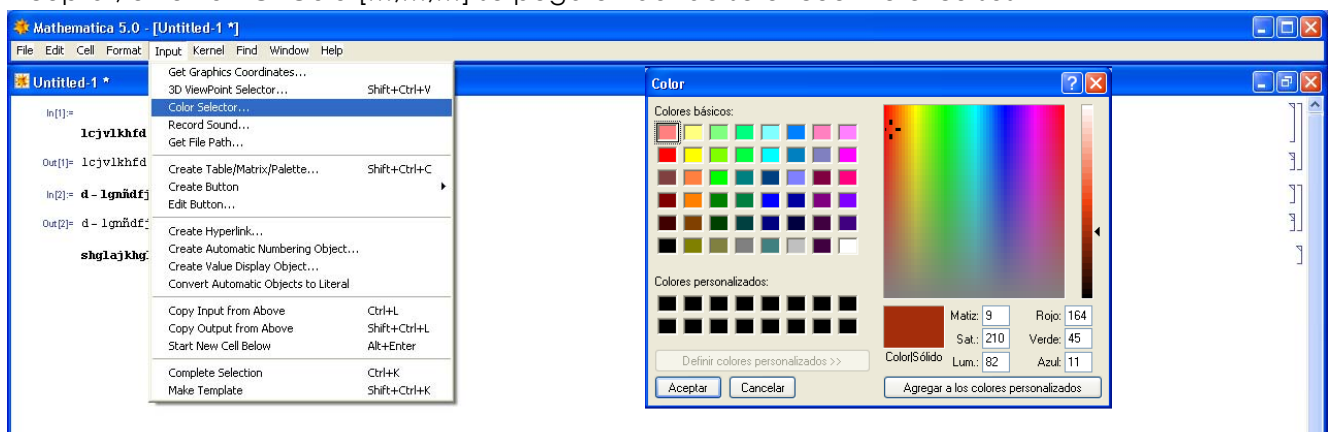
Si se coloca la instrucción **??** seguida del nombre del comando se pueden apreciar el listado de opciones del **Plot**:

### **??Plot**

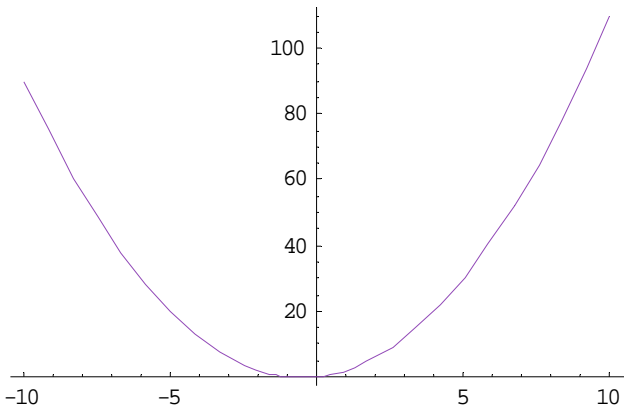
Plot[f, {x, xmin, xmax}] generates a plot of f as a function of x from xmin to xmax. Plot[{f1, f2, ...}, {x, xmin, xmax}] plots several functions fi. [More...](#)

```
Options [Plot] = {AspectRatio ->  $\frac{1}{\text{GoldenRatio}}$ , Axes -> Automatic, AxesLabel -> None,
  AxesOrigin -> Automatic, AxesStyle -> Automatic, Background -> Automatic,
  ColorOutput -> Automatic, Compiled -> True, DefaultColor -> Automatic,
  DefaultFont -> $DefaultFont, DisplayFunction -> $DisplayFunction, Epilog -> {},
  FormatType -> $FormatType, Frame -> False, FrameLabel -> None, FrameStyle -> Automatic,
  FrameTicks -> Automatic, GridLines -> None, ImageSize -> Automatic, MaxBend -> 10.,
  PlotDivision -> 30., PlotLabel -> None, PlotPoints -> 25, PlotRange -> Automatic,
  PlotRegion -> Automatic, PlotStyle -> Automatic, Prolog -> {}, RotateLabel -> True,
  TextStyle -> $TextStyle, Ticks -> Automatic }
```

**Añadir color a la función en v. 5:** Comando **PlotStyle**→Se puede utilizar un comando de color, como por ejemplo el **RGBColor[nro, nro, nro]** entre los corchetes van tres números reales entre 0 y 1 que indican los niveles de rojo verde y azul en la combinación. En la barra de menús en **Input->ColorSelector** se encuentra una ventana de colores para insertar el comando. Cuando se clickea Aceptar, el texto **RGBColor[.....]** se pega en donde se encuentre el cursos







En v. 6 y 7: Los colores ya están incorporados, solamente se debe escribir el nombre (en inglés y con la primer letra en mayúscula) después de la opción **PlotStyle**.

Si no se sabe el nombre de un color puede encontrárselo en el menú **Palettes**→**ColorScheme**, aparece una ventana de colores, en donde hay combinaciones y espectros de todo tipo; en la opción **Named** hay 6 listas de colores, clasificados por tema (geografía, química, etc). Luego de elegir el color deseado, se sitúa el cursor al lado de la → del PlotStyle y se hace click en **Insert** en la ventana ColorScheme. En el ejemplo se seleccionó el color rosa y luego de hacer click en Insert se escribió Pink en la instrucción del Plot. La función graficada sale rosa.

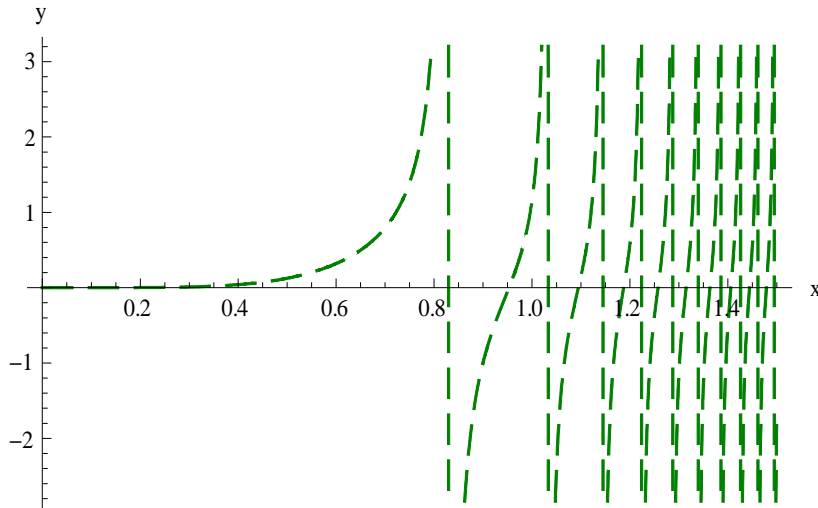
En las gráficas a continuación se han utilizado las siguientes opciones:

**PlotStyle** para el color, grosor y tipo de línea de la función, **GrayLevel** significa “Nivel de Gris”, **Thickness** es el grosor de la línea, y **Dashing** tipo de una línea. **AxesLabel** para titular los ejes, **Background** colorea el fondo.

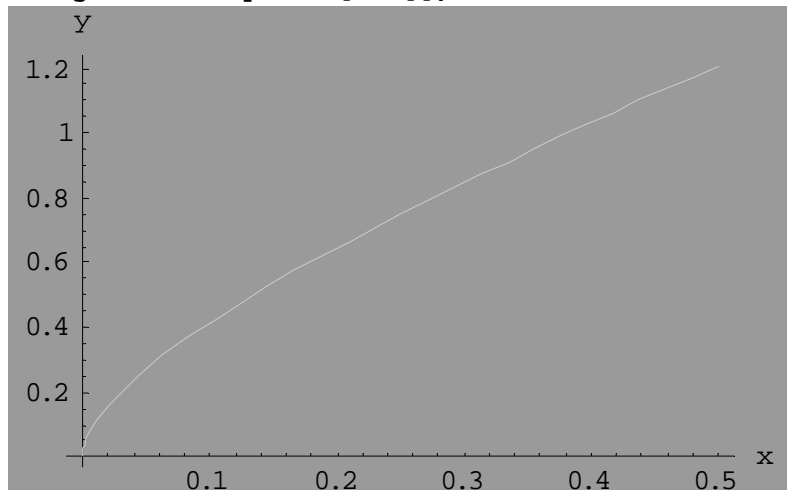
Toda opción con la palabra “**Style**” indica formato, y con la palabra “**Label**” indica título. La instrucción queda de la forma siguiente:



```
Plot[Tan[4 x^5],{x,0,1.5},PlotStyle->{RGBColor[0,0.501961,0],
Dashing[{0.04,0.02}]}, AxesLabel->{"x","y"}]
```



```
Plot[j[x],{x,0,0.5},PlotStyle->GrayLevel[0.6],AxesLabel->{"x","y"},
Background->GrayLevel[0.2]];
```



**Nota muy importante:** Una gran diferencia entre las versión 5 y las 6 y 7 es en la salida resultante a la ejecución de un comando Plot. En la versión 5, el colocar un punto y coma (;) al final de la instrucción como se ve en los ejemplos evita una salida posterior a la gráfica generada de una pequeña leyenda que dice "Graphics". Para evitar la salida de la gráfica después de ejecutada la instrucción (como por ejemplo en el caso de generar varias gráficas intermedias para lograr una final completa y querer que solo se vea la gráfica final y no todas las intermedias) se utiliza dentro de los corchetes del Plot la opción **DisplayFunction->Identity**, esto inhabilita la salida de la gráfica. Posteriormente se usará una opción que contrarresta al **DisplayFunction->Identity**, como después se verá.



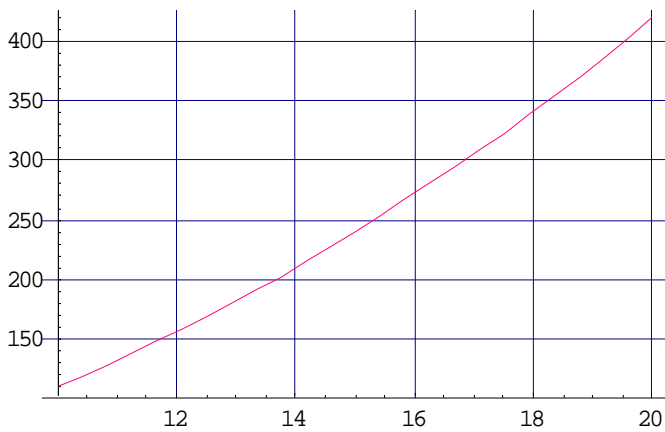
En las versiones 6 y 7 para evitar la salida en pantalla de la gráfica al ejecutar una instrucción con **Plot** simplemente se coloca un punto y coma (;) al final de la instrucción.

Por eso si Ud. está copiando una instrucción de un archivo realizado en el Mathematica 5, **elimine** todos los (;) porque sino no va a ver ninguna gráfica.

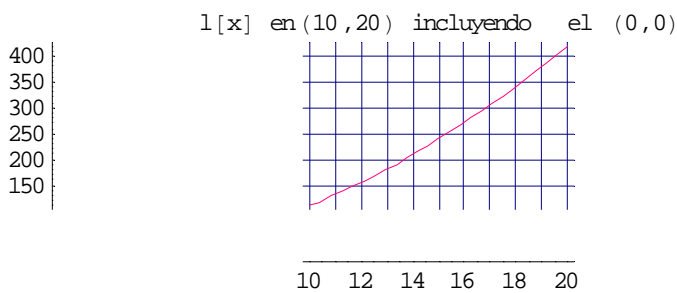
### Otras opciones

Cuando el rango de valores de la variable independiente está lejos del origen de coordenadas, el Mathematica corre el mismo en la graficación mostrando la gráfica en un rango de graficación apropiado para su visualización. Sin embargo, con los opciones del comando Plot **AxesOrigin** y **PlotRange** se puede forzar el gráfico para que incluya el origen y muestre los ejes en toda su dimensión. La opción **GridLines** muestra un grillado en el área de la gráfica, con posibilidad de definir el entramado.

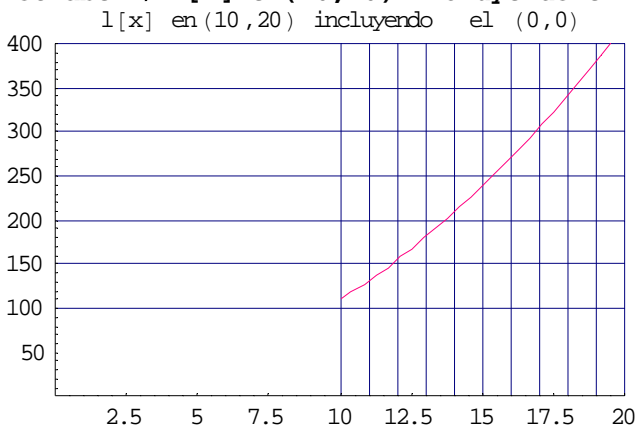
```
Plot[p[x],{x,10,20},PlotStyle→RGBColor[1, 0, 0.501961],GridLines→Automatic];
```



```
Plot[p[x],{x,10,20},PlotStyle→RGBColor[1, 0, .501961],
GridLines→{Range[10,20,1],Range[100,400,50]},AxesOrigin→{0,0},
PlotLabel→"l[x] en(10,20) incluyendo el (0,0)"];
```



```
Plot[p[x],{x,10,20},PlotStyle→RGBColor[1, 0, 0.501961],
GridLines→{Range[10,20,1],Range[100,400,50]},AxesOrigin→{0,0},
PlotLabel→"l[x] en(10,20) incluyendo el (0,0)",PlotRange→{{0,20},{0,400}}];
```

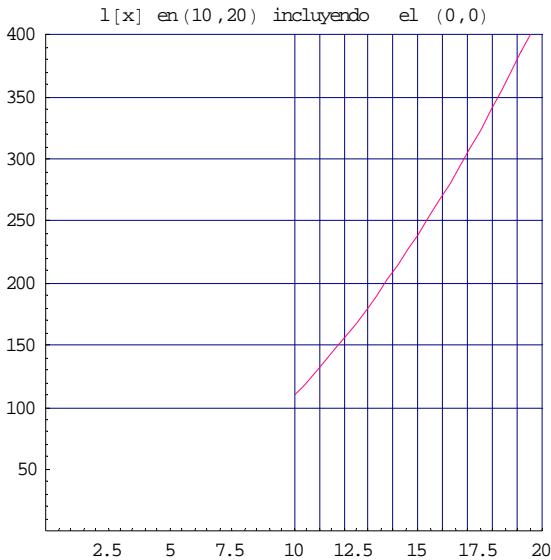


**Nota:** La valoración o asignación de valores o lista de valores de una opción del comando Plot se hace a través de la  $\rightarrow$ .

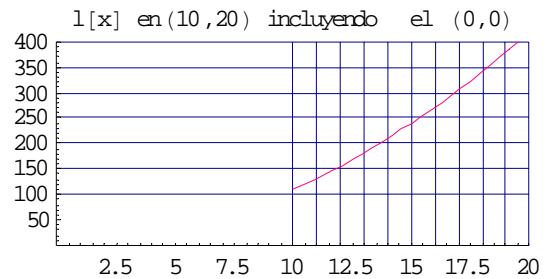
En estos casos, queda evidenciado que la escala de los ejes no es 1 a 1, sino que el programa se adapta para poder exhibir una gráfica de visualización aceptable. El parámetro **AspectRatio** se utiliza para forzar una escala específica. El valor **Automatic** de este parámetro indica una escala 1 a

1, aunque si la gráfica queda desproporcionada con respecto a la hoja de trabajo, el Mathematica se aparta de dicha escala.

```
Plot[p[x],{x,10,20},PlotStyle->RGBColor[1, 0, 0.501961],GridLines->{Range[10,20,1],
Range[100,400,50]},AxesOrigin->{0,0},PlotLabel->"l[x] en(10,20) incluyendo
el (0,0)",PlotRange->{{0,20},{0,400}},AspectRatio->1];
```



Y con AspectRatio-> 0.4



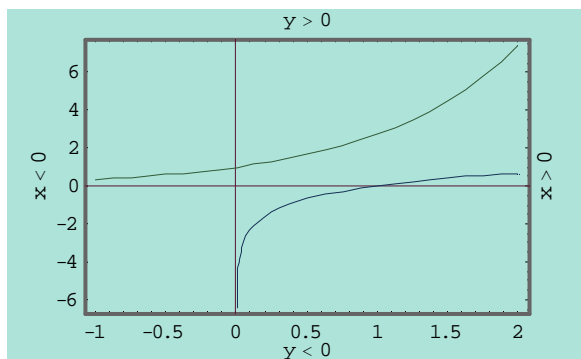
Las combinaciones son infinitas, y una buena elección de opciones puede generar gráficas claras y de gran calidad.

A veces, la excesiva cantidad de detalles hacen dificultosa la visión de la gráfica, pero la misma puede ser redimensionada. Si se la clickea con el mouse, se puede modificar su tamaño. También si se presiona la tecla **Ctrl** mientras está seleccionada la gráfica, entonces se pueden observar las coordenadas dentro de la gráfica en donde está posicionado el Mouse, y se pueden marcar puntos.

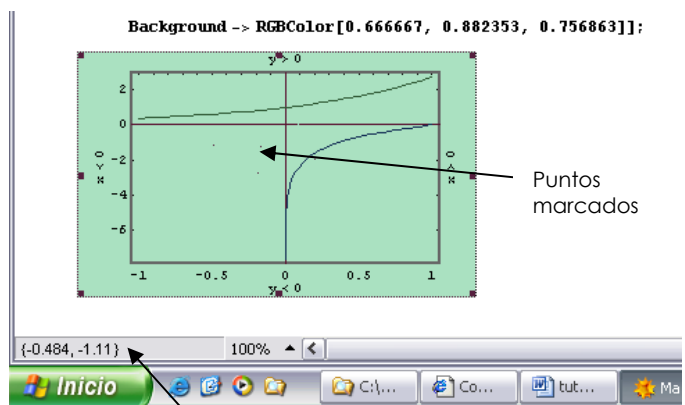
```
z[x_]= Log[x];
```

```
w[x_]= Exp[x];
```

```
Plot[{z[x],w[x]},{x,-1,1},PlotStyle->{RGBColor[0.0823529, 0.184314, 0.333333],
RGBColor[0.188235, 0.352941, 0.211765]},Frame->True,FrameStyle->{GrayLevel[0.4],
Thickness[0.01]},FrameLabel->{y < 0,x < 0,y > 0,x > 0}, AxesStyle->RGBColor[0.34902,
0.0666667, 0.188235],Ticks->{{-1,0.5,0,0.5},{-6,-5,-4,-3,-2,-1,0,1,2}},
Background->RGBColor[0.666667, 0.882353, 0.756863]];
```



Ojo: SOLO en Mathematica hasta versión 5

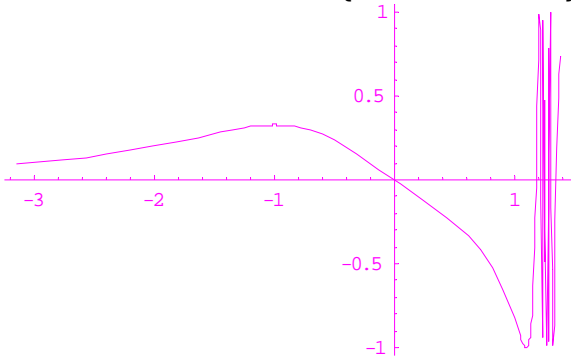


Coordenadas que indican la posición del Mouse en la gráfica

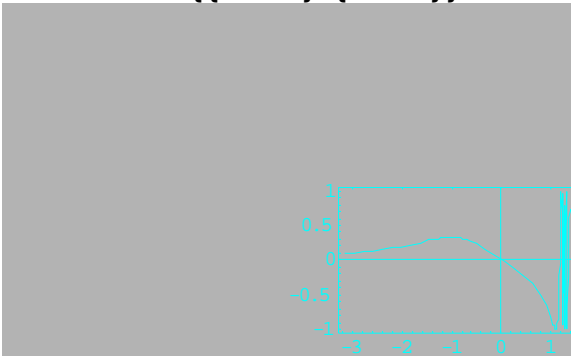
**Nota:** Aquí se muestra la gráfica de dos funciones, las mismas están escritas en forma de **Lista** entre llaves y separadas por comas. La opción **Frame** produce un marco que rodea la gráfica.

Si se desea generar una gráfica descentrada dentro de la misma región de ploteo existe el parámetro **PlotRegion**, que al darle valores entre 0 y 1 para los ejes, proporciona una forma de controlar el tamaño y la ubicación de la salida gráfica.

```
Plot[Sin [x/(x^3-2)],{x,-π, 7/16 π},DefaultColor→RGBColor[1,0,1];
```



```
Plot[Sin[x/(x^3-2)],{x,-π, 7/16 π}, DefaultColor→RGBColor[0,1,1],
PlotRegion→{{0.5,1},{0,0.5}},Frame→True, Background→GrayLevel[0.7];
```



**Nota:** En este Plot la función no está predefinida sino que se ingresa directamente, esto se usa cuando solo interesa su graficación.

En los valores del dominio de graficación, se utilizó del **Palette** → **BasicInput** el símbolo del número Pi.



El parámetro **DefaultColor** se utiliza en la versión 5 e indica el color de salida de toda la gráfica incluyendo ejes, marco, etc. En las versiones 6 y 7 este parámetro es reemplazado por **BaseStyle** y se utiliza exactamente igual

El palette **BasicInput** en v. 6 se llama **BasicMathInput**.

Parámetros actualizados y optimizados en v. 6: **PlotRange**, **AxesLabel**, **AxesStyle**, **AspectRatio**, **Frame**, **FrameLabel**, **FrameStyle**. Se pueden ver estas modificaciones en el Help, pero su utilización básica sigue siendo la misma.

### Listas

En el Mathematica, una lista es un conjunto de elementos definido entre llaves y separados cada uno del otro por una coma.

```
lista={a,b,c,d,e,f,g,h,i,j}
```

```
{a,b,c,d,e,f,g,h,i,j}
```

Una lista puede contener a su vez listas como elementos, este anidamiento permite el ingreso de matrices como listas de listas, para su manipulación. Asimismo, una lista común puede interpretarse como un vector.

```
listaanidada= {{a,b,c},{m,n,f},{1,2},p,q,4,5,6,r}
```

```
{{a,b,c},{m,n,f},{1,2},p,q,4,5,6,r}
```

```
%//MatrixForm
```

$$\begin{pmatrix} \{a, b, c\} \\ \{m, n, f\} \\ \{1, 2\} \\ p \\ q \\ 4 \\ 5 \\ 6 \\ r \end{pmatrix}$$

**Nota:** La sentencia "%//MatrixForm" muestra un conjunto dentro de una lista en forma más prolija y de fácil visibilidad y comprensión. El prefijo % indica que se refiere a la instrucción inmediata anterior **ingresada**, y el // es una forma rápida de aplicar un comando a una instrucción ya escrita, es el equivalente de **MatrixForm[%]** generando la misma salida.

La generación de listas es una herramienta muy importante, para ello existe el comando **Table**. La instrucción posee dos argumentos: uno en donde se genera el elemento según una expresión, y el segundo es un rango de valores del "contador" de elementos, que indica cuántas veces se aplicará la expresión del primer argumento. El contador también puede ser usado como parte de la expresión del primer argumento.

```
Table[Range[j],{j,5}]
{{1},{1,2},{1,2,3},{1,2,3,4},{1,2,3,4,5}}
Table[x^i - i^2,{i,1,10}]
{-1+x, -4+x^2, -9+x^3, -16+x^4, -25+x^5, -36+x^6, -49+x^7, -64+x^8, -81+x^9, -100+x^10}
```

```
Table[(n-IntegerPart[n])/n,{n,10,1,-0.9}]
{0,0.010989,0.0243902,0.0410959,0.0625,0.0909091,0.130435,0.189189,0.285714,0.473684,1.}
```

**Nota:** **Range** crea una lista de elementos empezando de 1 hasta el valor del parámetro. **IntegerPart** obtiene la parte entera de un valor. En el segundo argumento de la última lista, el contador va de 10 a 1 con saltos de -0.9, o sea que va disminuyendo.

También se puede construir una tabla (o lista) de gráficos con variaciones. Esto es útil, por ejemplo, para observar el impacto de un parámetro en una función.

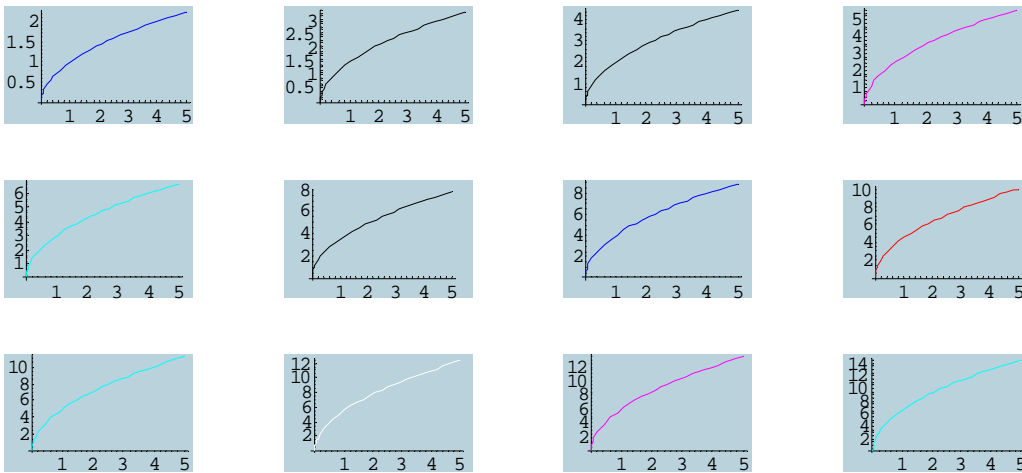
```
graficas=Table[ a[h]=Plot[ h Sqrt[x],{x,0,5},PlotStyle->RGBColor[Random[Integer],
Random[Integer],Random[Integer]],Background->RGBColor[0.72549, 0.823529,0.858824],
DisplayFunction->Identity],{h,1,6,0.5}];
```

El incremento de h se hace con saltos de 0.5, ese h modifica la gráfica porque está multiplicando a la función Raíz cuadrada. Se generan 12 gráficas, donde la función va tomando valores cada vez más altos para un mismo valor de la variable independiente.

La instrucción que fue nombrada "**graficas**" no genera ninguna salida debido a que Plot incluye la sentencia **DisplayFunction->Identity**. (para v. 5)

El comando **Show** puede mostrar diferentes gráficas generadas anteriormente, llamándolas si las mismas han sido nombradas o con la utilización del "%" con la utilización del comando **GraphicsArray** que permite la visualización de una matriz de gráficos. En este caso la matriz fue construida por el comando **Partition**, que divide el vector *graficas* en una matriz de tres filas por cuatro columnas. El uso del **Show[GraphicsArray[.....]** elimina las llaves de la partición.

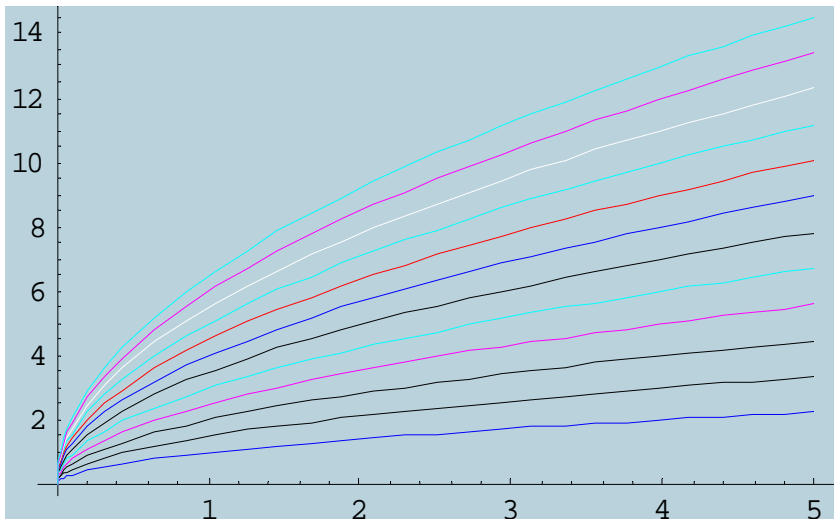
```
Partition[graficas,4];
Show[GraphicsArray[%],GraphicsSpacing->.5, DisplayFunction->isplayFunction]];
```



En la v. 6 y 7 la opción **GraphicsArray** no existe ni sus opciones o parámetros, se utiliza directamente en vez del Show el comando **GraphicsGrid** con todas sus opciones.

A su vez, las diferencias entre las gráficas son más observables cuando todas son graficadas dentro de un mismo frame de graficación.

**Show[graficas,DisplayFunction→\$DisplayFunction];**



Recordar que en la v. 6 y 7 el parámetro **DisplayFunction** no existe, solamente se debe colocar puntos y comas al fin de cada instrucción y NO en el Show:

**Partition[graficas,4];**  
**GraphicsGrid [%]**

### Puntos

Todo par de números reales ordenado en la forma (x,y) expresan las coordenadas de un *punto* y su posición en el plano cartesiano. Si en el Mathematica armamos una lista de pares de números, estamos ingresando un conjunto de puntos que pueden ser graficados en el plano.

**punto={0.3,0.6}**  
 {0.3,0.6}

**listapuntos={{0.3,0.6},{-3,-7}, {0.233,1.56}, {3.78,0.11}, {2,2}, {9.24,5.88},**  
**{0,0},{1,7},{4,1}}**  
 {{0.3,0.6},{-3,-7},{0.233,1.56},{3.78,0.11},{2,2},{9.24,5.88},{0,0},{1,7},{4,1}}

Algunos comandos para la manipulación de listas son:

Ordering: muestra la posición de cada elemento según el orden a que pertenece.

**Ordering[listapuntos]**

{2, 7, 3, 1, 8, 5, 4, 9, 6}

Sort: ordena los elementos

**Sort[listapuntos]**

{{-3, -7}, {0, 0}, {0.233, 1.56}, {0.3, 0.6}, {1, 7}, {2, 2}, {3.78, 0.11}, {4, 1}, {9.24, 5.88}}

Max - Min: muestra el mayor - menor elemento de la lista, incluyendo sublistas. En nuestro ejemplo, cada punto es una sublista, por lo tanto estos comandos mostrarán la mayor - menor coordenada.

**Max[listapuntos]**

9.24

**Min[listapuntos]**

-7

Length: cantidad de elementos

**Length[listapuntos]**

9

Dimensions: indica no solo la cantidad de elementos, también la cantidad de subelementos que forman cada elemento, pero si alguno tiene un formato diferente, Dimensions solamente indicará la cantidad de elementos tal como el comando Length

**Dimensions[listapuntos]**

{9, 2}

Append, Prepend e Insert agregan elementos en diferentes lugares de la lista:

**l=Append[listapuntos, {1, 2, 3}]**

{{0.3, 0.6}, {-3, -7}, {0.233, 1.56}, {3.78, 0.11}, {2, 2}, {9.24, 5.88}, {0, 0}, {1, 7}, {4, 1}, {1, 2, 3}}

**Dimensions[l]**

{10}

**Prepend[listapuntos, {-1, -1}]**

{{-1, -1}, {0.3, 0.6}, {-3, -7}, {0.233, 1.56}, {3.78, 0.11}, {2, 2}, {9.24, 5.88}, {0, 0}, {1, 7}, {4, 1}}

**Insert[listapuntos, {E, E^2}, {{1}, {2}, {-1}, {-2}}]**

{{e, e^2}, {0.3, 0.6}, {e, e^2}, {-3, -7}, {0.233, 1.56}, {3.78, 0.11}, {2, 2}, {9.24, 5.88}, {0, 0}, {1, 7}, {e, e^2}, {4, 1}, {e, e^2}}

Otros comandos útiles para la formación y manipulación de listas pueden verse en la Ayuda del Mathematica.

Una forma rápida de construir listas de puntos es con el comando Table:

**Clear[f]**

$$f[x_] = \frac{x^3 + 2}{x^4 - 1}$$

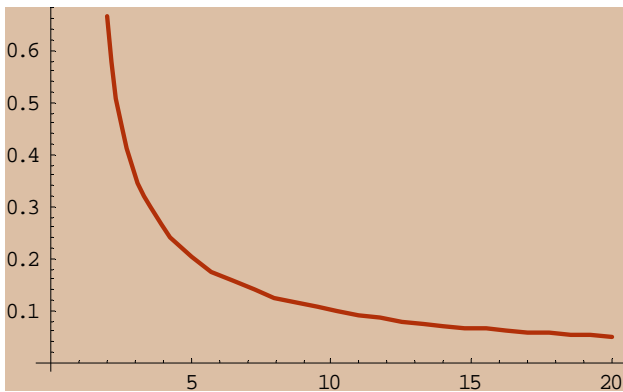
**puntosf=Table[{x, f[x]}, {x, 2, 20, 1}]/N**

{{2., 0.666667}, {3., 0.3625}, {4., 0.258824}, {5., 0.203526}, {6., 0.16834}, {7., 0.14375}, {8., 0.125519}, {9., 0.111433}, {10., 0.10021}, {11., 0.0910519}, {12., 0.0834338}, {13., 0.0769958}, {14., 0.0714825}, {15., 0.0667075}, {16., 0.0625315}, {17., 0.0588482}, {18., 0.0555751}, {19., 0.0526473}, {20., 0.0500128}}

**Nota:** la función fue ingresada en forma gráfica con la utilización del Palette **BasicInput** (en v. 5)

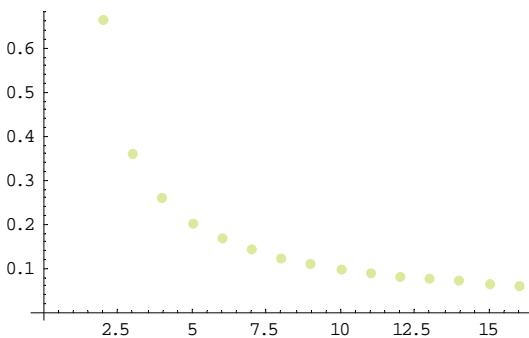
**plotf=Plot[f[x], {x, 2, 20}, PlotStyle->{RGBColor[0.694118, 0.188235, 0.0392157], Thickness[0.01]}, Background->RGBColor[0.862745, 0.74902, 0.647059]];**



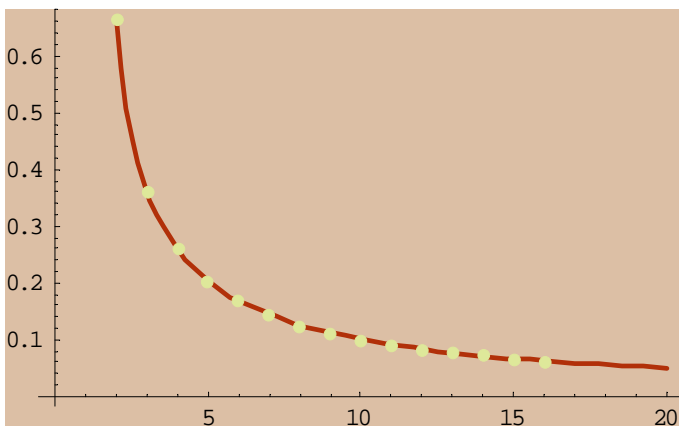


Ahora graficaremos los puntos con el comando **ListPlot**:

```
plotpuntos=ListPlot[puntosf,PlotStyle->{RGBColor[0.870588, 0.909804, 0.6],
PointSize[0.02]}];
```



```
Show[plotf,plotpuntos];
```



Para evitar la generación parcial de gráficas, se puede usar el parámetro **DisplayFunction**, explicado anteriormente (para versión 5).

### Tablas

También es útil la valoración de funciones en las cercanías de ciertos puntos, para observar tendencias y sacar conclusiones acerca de ciertas zonas específicas, como puntos de discontinuidad, o asíntotas, etc.

Por ej la función  $f(x)$  no existe en el -1 ni en el 1 debido a que el denominador se hace cero. Una tabla de valores cercanos a esos puntos indica el comportamiento de la función.

```
tendenciasneg=Table[{x,f[x]},{x,-2,-0.1,0.1}]
tendenciasposit=Table[{x,f[x]},{x,0,1.9,0.1}]
```

```
Power::infty : Infinite expression  $\frac{1}{0}$  encountered . More..
```

```
{{-2.0000,-0.4},{-1.9,-0.403836},{-1.8,-0.40347},{-1.7,-0.396213},{-1.6,-0.377413},{-1.5,-0.338462},{-1.4,-0.261824},{-1.3,-0.106137},{-1.2,0.253353},{-1.1,1.4415},{-1.,ComplexInfinity},{-0.9,-3.69584},{-0.8,-2.52033},{-0.7,-2.18055},{-0.6,-2.04963},{-0.5,-2.},{-0.4,-1.98686},{-0.3,-1.98911},{-0.2,-1.99519},{-0.1,-1.9992}}
```

```
Power::infy : Infinite expression  $\frac{1}{0}$  encountered . More..
```

```
{{0,-2},{0.1,-2.0012},{0.2,-2.01122},{0.3,-2.04355},{0.4,-2.11823},{0.5,-2.26667},{0.6,-2.54596},{0.7,-3.0833},{0.8,-4.25474},{0.9,-7.93545},{1.,ComplexInfinity},{1.1,7.17733},{1.2,3.47243},{1.3,2.26119},{1.4,1.66948},{1.5,1.32308},{1.6,1.09767},{1.7,0.940276},{1.8,0.824629},{1.9,0.73628}}
```

**Dimensions[tendenciasneg]**

```
{20,2}
```

Las tablas creadas están formadas por 20 elementos, cada uno de ellos cuenta con dos subelementos. Para indicar un elemento o subelemento de una lista, el Mathematica utiliza la siguiente notación:

**tendenciasneg[[11]]**

```
{-1.,ComplexInfinity}
```

Es el elemento nº 11 de la lista tendenciasneg, es el par (-1, ComplexInfinity), en donde la palabra ComplexInfinity indica la indeterminación.

**tendenciasposit[[11,2]]**

```
ComplexInfinity
```

Es el 2º subelemento del elemento nº 11 de la lista tendenciasposit, es la expresión ComplexInfinity, que indica que hay una discontinuidad en ese punto.

**TableForm[{{TableForm[tendenciasneg]},TableForm[tendenciasposit]}],  
TableSpacing->{1,1}]**

-2	$-\frac{2}{5}$	0	-2
-1.9	-0.403836	0.1	-2.0012
-1.8	-0.40347	0.2	-2.01122
-1.7	-0.396213	0.3	-2.04355
-1.6	-0.377413	0.4	-2.11823
-1.5	-0.338462	0.5	-2.26667
-1.4	-0.261824	0.6	-2.54596
-1.3	-0.106137	0.7	-3.0833
-1.2	0.253353	0.8	-4.25474
-1.1	1.4415	0.9	-7.93545
-1.	ComplexInfinity	1.	ComplexInfinity
-0.9	-3.69584	1.1	7.17733
-0.8	-2.52033	1.2	3.47243
-0.7	-2.18055	1.3	2.26119
-0.6	-2.04963	1.4	1.66948
-0.5	-2.	1.5	1.32308
-0.4	-1.98686	1.6	1.09767
-0.3	-1.98911	1.7	0.940276
-0.2	-1.99519	1.8	0.824629
-0.1	-1.9992	1.9	0.73628

El comando **TableForm** proporciona un formato de salida apropiado para la visualización de datos. En la instrucción utilizamos el comando en forma anidada para lograr un posicionamiento paralelo de las listas, como también el uso de los juegos de llaves dobles, el Mathematica considera de esa manera al conjunto como una matriz de una sola fila, y cada elemento es una de las lista en forma de columnas.

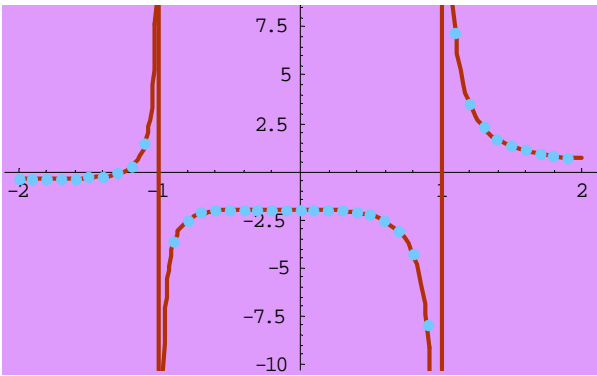
**tendencias=Join[Delete[tendenciasneg,11],Delete[tendenciasposit,11]]**

```
{{-2, -2/5}, {-1.9, -0.403836}, {-1.8, -0.40347}, {-1.7, -0.396213}, {-1.6, -0.377413}, {-1.5, -0.338462}, {-1.4, -0.261824},
{-1.3, -0.106137}, {-1.2, 0.253353}, {-1.1, 1.4415}, {-0.9, -3.69584}, {-0.8, -2.52033}, {-0.7, -2.18055}, {-0.6, -2.04963},
{-0.5, -2.}, {-0.4, -1.98686}, {-0.3, -1.98911}, {-0.2, -1.99519}, {-0.1, -1.9992}, {0, -2}, {0.1, -2.0012}, {0.2, -2.01122},
{0.3, -2.04355}, {0.4, -2.11823}, {0.5, -2.26667}, {0.6, -2.54596}, {0.7, -3.0833}, {0.8, -4.25474}, {0.9, -7.93545}, {1.1, 7.17733},
{1.2, 3.47243}, {1.3, 2.26119}, {1.4, 1.66948}, {1.5, 1.32308}, {1.6, 1.09767}, {1.7, 0.940276}, {1.8, 0.824629}, {1.9, 0.73628}}
```

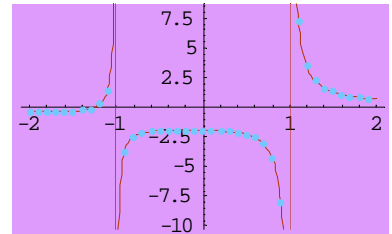
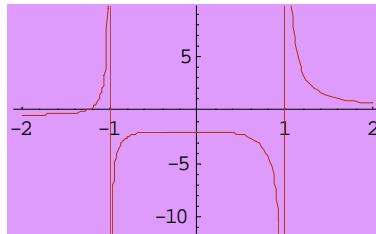
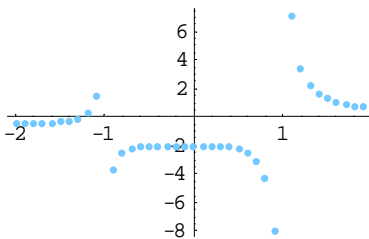
```
ten=ListPlot[tendencias,PlotStyle->{RGBColor[0.45098, 0.0705882, 0.596078],
PointSize[0.02]},DisplayFunction->Identity];
```

```
plotf2=Plot[f[x],{x,-2,2}, PlotStyle->{RGBColor[0.694118, 0.188235,
0.0392157],Thickness[0.01]},Background>Hue[.78383,.3838383,.98437],
DisplayFunction->Identity];
```

```
Show[plotf2,ten, DisplayFunction->$DisplayFunction];
```

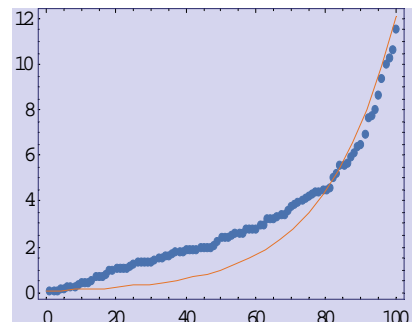
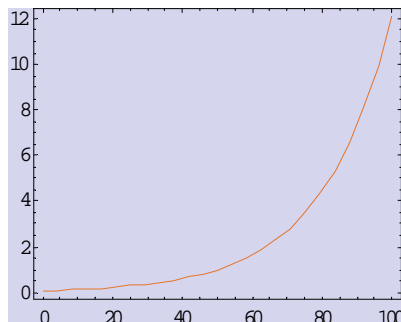
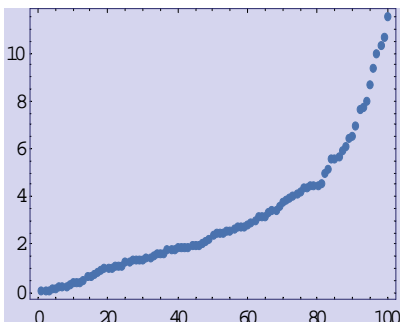


```
Show[GraphicsArray[{ten,plotf2,t},GraphicsSpacing->.2, DisplayFunction->
$DisplayFunction]];
```



La graficación de puntos es interesante cuando éstos son datos tomados de un proceso o sistema real, como una muestra de valores que toma un instrumento de medición o alguna característica de una población, etc.

Se ingresan los datos obtenidos y su graficación permite un estudio de la posible tendencia que tiene la variable en estudio. La siguiente gráfica, construida con los mismos comandos utilizados anteriormente, muestra un pequeño análisis de la cantidad de microbios presentes en un producto según la cercanía a su fecha de vencimiento, luego se aplica una aproximación.





En la versión 6 los comandos **Table** y **ListPlot** ha sido modificados, el **ListPlot** en muchas opciones y le han sido agregadas otras muy interesantes. Estas opciones son para personalizar y optimizar los gráficos. Pueden observarlas es en el **Help**, en donde se explican detalladamente.

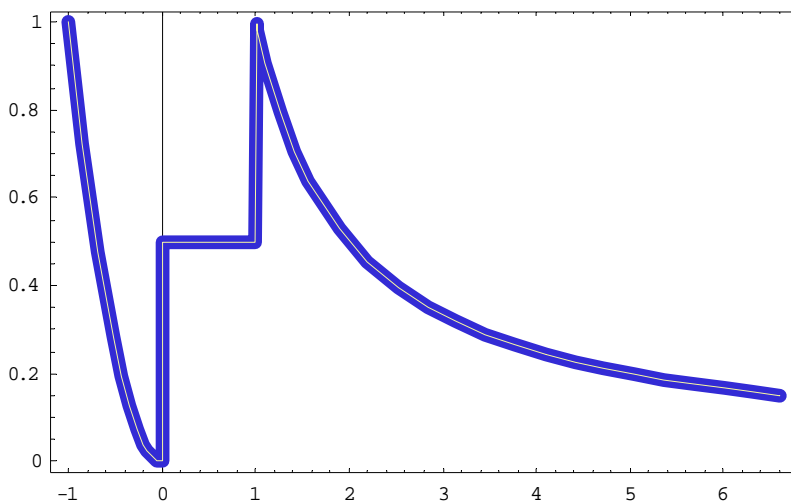
### Definición de funciones por tramos

El Mathematica permite la definición de funciones por tramos de dos formas, una por definición y otra por el uso de comandos.

```
tramosd[x_]:=x^2;/x<0;
tramosd[x_]:=0.5 /; 0 <= x <= 1;
tramosd[x_]:= 1/x /; x>1
```

```
tramosc[x]:= Which[x < 0, x^2, x > 1, 1/x, True, 0.5];
```

```
Plot[{tramosd[x],tramosc[x]},{x,-1,6.6},PlotStyle->{{RGBColor[0.2, 0.168627,
0.835294],Thickness[0.02]}, RGBColor[0.976471, 0.968627, 0.635294]},
Frame->True];
```



Observamos que **tramosd** y **tramosc** son la misma función. La sintaxis de la definición por tramos tiene dos partes, primero se escribe la ley y luego el dominio de definición, separados por un **/;** que significa "si". Es importante notar que cuando se define, se deben colocar los dos puntos antes del igual (**:=**), porque sino el kernel, evalúa las instrucciones línea por línea como un lenguaje intérprete, genera una salida de error.

En el comando **Which** incluye cada definición separada por coma e inmediatamente el dominio de definición, si el dominio no cubre todos los reales, la opción **True** obliga a la función a tomar el valor descripto a continuación, en el ejemplo es **0.5**, se ve que entre 0 y 1 la función no está definida por lo tanto en la gráfica la misma vale=0.5.

### Nota Final

Estas primeras indicaciones son las guías esenciales para abordar las propuestas de los primeros trabajos prácticos, de las asignaturas del Área Matemática de 1º año de Ingeniería.

Este Manual es un documento abierto, se incorporará gradualmente durante el año las instrucciones necesarias para lograr el aprendizaje apropiado del software, y por consiguiente, adquirir las destrezas y habilidades requeridas para trabajar en forma autónoma durante el proceso de elaboración de los trabajos.

### Links importantes

A través de estos links se puede acceder al universo del Mathematica:

Sitio Principal: <http://www.wolfram.com/>

MathWorld (Mathematics Resources): <http://mathworld.wolfram.com/>

Wolfram Alpha (Computational Knowledge Engine): <http://www.wolframalpha.com/>